

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
17 May 2001 (17.05.2001)

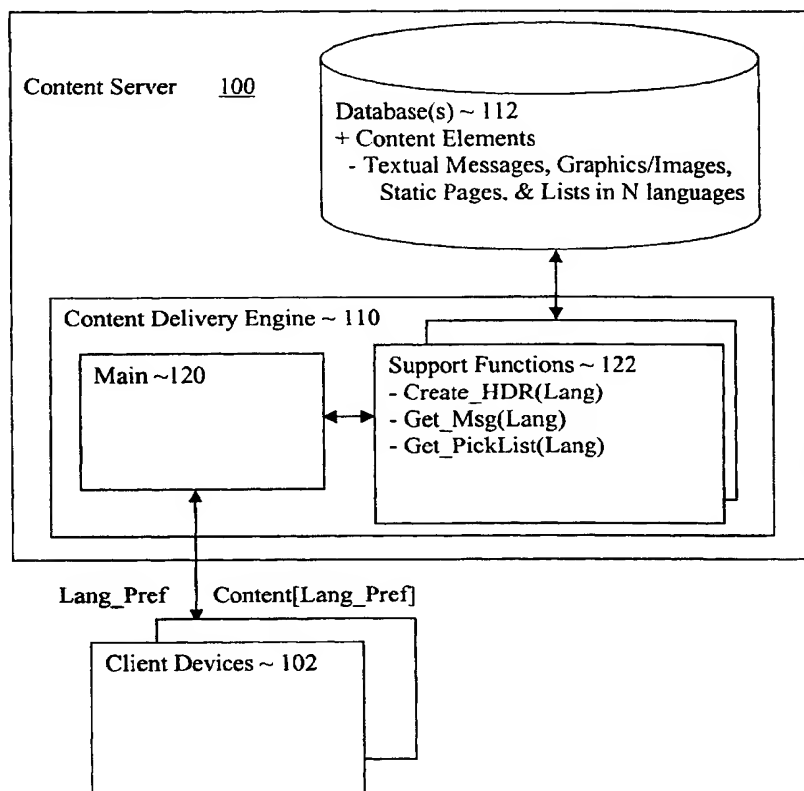
PCT

(10) International Publication Number  
**WO 01/35215 A2**

- (51) International Patent Classification<sup>7</sup>: **G06F 9/44** (74) Agents: AUYEUNG, Aloysius, T., C. et al.; Columbia IP Law Group, LLC, Suite 109, 4900 SW Meadows Road, Lake Oswego, OR 97035 (US).
- (21) International Application Number: PCT/US00/30646
- (22) International Filing Date:  
6 November 2000 (06.11.2000) (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
60/164,585 10 November 1999 (10.11.1999) US
- (71) Applicant (*for all designated States except US*): GLOBAL MARKET INSITE, INC. [US/US]; 1474 NE Katsura Street, Issaquah, WA 98029 (US).
- (71) Applicant and (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- (72) Inventor: MONSTER, Robert, W. [US/US]; 1565 Jonquil Lane, Issaquah, WA 98029 (US).

[Continued on next page]

(54) Title: CONTENT DELIVERY IN A PREFERRED LANGUAGE FOR A LARGE NUMBER OF LANGUAGES



(57) Abstract: A content delivery engine is provided with a number of support functions to support content delivery in a preferred language selected from a large number of supported languages. In one embodiment, the functions include a first function to set a language as a preferred language, a second function to set a character code setting of a content page based on the preferred language, a third function to generate graphic/image contents of the content page, a fourth function to generate textual contents of the content page, a fifth function to generate one or more links expressed in the preferred language for the content page, a sixth function to generate pick lists for the content page, ordered in view of the preferred language, and a seventh function to generate a drop down list of languages for the content page for changing the preferred language to another one of the supported languages. In alternate embodiments, more or less of these preferred language based content delivery support functions are included.



WO 01/35215 A2



**Published:**

— Without international search report and to be republished upon receipt of that report.

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## **Content Delivery In A Preferred Language For A Large Number Of Languages**

### Related Application

This application claims priority to U.S. Provisional Application number 09/164,585, entitled "System and Method for Obtaining and Collating Survey Information In Real Time For Multiple Languages and Multiple Character Encodings", filed on November 10, 1999, which is hereby fully incorporated by reference.

### BACKGROUND OF THE INVENTION

#### 1. Field of the Invention

The present invention relates to the field of data processing. More specifically, the present invention relates to content delivery in a preferred language of a user in a computing environment where a large number of languages are supported.

#### 2. Background Information

Historically, to provide better user experience, software vendors exporting their software products to non-English speaking regions would translate the messages presented in the user interfaces of the products into the various native languages of the regions, and correspondingly packaged the translated messages into the different versions shipped to the different regions. To facilitate the translation and packaging, typically the translated elements are isolated from the logic of the product, and the logic would reference the messages by their identifiers, thereby eliminating the need for the logic to be cognizant of the language setting the product is executing in, and the translators would only have to work with elements to be translated. For example, if the logic is to display a particular message on certain condition, the logic would simply reference the message by its identifier, such as a message number, without caring whether it is a French version or a Japanese version of

the message it wants. By virtue of proper packaging, i.e. having the messages expressed in French packaged in the French version, and the messages expressed in Japanese packaged in the Japanese version, and so forth, the correct version of the translated message would be displayed for the user.

This prior art approach is often referred to as localization of products, and localization has worked well for decades, in the old “unconnected” computing environment, where software vendors can safely assume that a French version of the product, by virtue of its controlled distribution, will be executed in a French speaking region, and French would most likely be the preferred language of the user. Likewise, a Japanese version of the product, also by virtue of its distribution, will be executed in a Japanese-speaking region, and Japanese would most likely be the preferred language of the user. Moreover, in each case, the underlying executing environment would be able to provide the proper character code set support, i.e. the French character set for the French version, the Japanese character set for the Japanese version and so forth.

However, with advances in integrated circuit, microprocessor, networking and communication technologies, increasing number of computing devices, are being interconnected together, through private and public networks, such as the Internet. The increased interconnectivity of computing devices have led to wide spread adoption of various network dependent applications, such as email, the world wide web, e-commerce and various application services offered by a wide range of application service providers.

Similar to the software exporters of the earlier era, to improve user experience, content site operators and application service providers have attempted to deliver content in the user's native language. Typically, a number of buttons labeled with the language supported or a number of flag icons representing the language supported, are displayed for the user to select a preferred language for the delivery of content. Upon selection, the appropriate version of the pre-translated and pre-packaged contents, would be provided to the user. The user is assumed to have the necessary character code set support for the selected language. This prior art button/icon and pre-packaging

approach is fine for as long as only a small number of the most widely supported languages are included, which was the case to-date. Due to affordability, private and public networks, such as the Internet, in substance mostly interconnect users in the most advanced developed countries, even though technologically, virtually any computing device from any part of the world can be interconnected together. Thus, content providers typically support only the half a dozen languages spoken in these most advanced developed countries, i.e. French, German, Japanese, Italian and so forth.

However, increasingly, as the relevant technologies continue to become even more affordable, and extend into other developed and developing countries, interconnection of computing devices have become even more global. As a result, content and service providers are expected to deliver contents and services to users of a much larger number of languages. Unlike the earlier years, where only a handful of non-English language are supported, sophisticated content/service providers are expected to support many more languages, from Scandinavian and Eastern Europe languages, to Middle East Arabic languages, as well as Chinese and other non-Japanese Asian languages, which can total to as much as 50-100 languages. One particular application that is taking the lead is consumer research or survey applications, where it often takes the lead in reaching the non-English speaking emerging markets. Research has consistently shown that it is vitally important to interview or survey the panelists in their native languages.

For various reasons, e.g., the amount of display area consumed by the language buttons/icons, the prior art techniques of offering buttons or flag icons to select a preferred language, and static pre-packaging of contents are no longer adequate in such a computing environment, where a large number of languages have to be supported. A more efficient and effective approach to delivering content in a user's preferred language for such computing environment is desired.

#### SUMMARY OF THE INVENTION

A content delivery engine is provided with a number of support functions to support content delivery in a preferred language selected from a large number of supported languages. In one embodiment, the functions include a first function to set a language as a preferred language, a second function to set a character code setting of a content page based on the preferred language, a third function to generate graphic/image contents of the content page, a fourth function to generate textual contents of the content page, a fifth function to generate one or more links expressed in the preferred language for the content page, a sixth function to generate pick lists for the content page, ordered in view of the preferred language, and a seventh function to generate a drop down list of languages for the content page for changing the preferred language to another one of the supported languages. In alternate embodiments, more or less of these preferred language based content delivery support functions are included.

#### BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references denote similar elements, and in which:

**Figure 1** illustrates an overview of the present invention, including the content delivery engine of the present invention, in accordance with one embodiment;

**Figure 2** illustrates a content view of a preferred language based delivered content, in accordance with one embodiment;

**Figure 3** illustrates the operational flow of the relevant aspects of the main module of **Fig. 1**, in accordance with one embodiment;

**Figure 4** illustrates the operational flow of the relevant aspects of the create header function of **Fig. 1**, in accordance with one embodiment;

**Figure 5** illustrates the operational flow of the main module of **Fig. 1** for getting language specific graphics/images, in further details, in accordance with one embodiment;

**Figure 6** illustrates the operational flow of the relevant aspects of the get message function of **Fig. 1**, in accordance with one embodiment;

**Figure 7** illustrates the operational flow of the main module of **Fig. 1** for generating links expressed in the preferred language, in further detail, in accordance with one embodiment;

**Figure 8** illustrates the operational flow of the relevant aspects of the get pick list function of **Fig. 1**, in accordance with one embodiment;

**Figure 9** illustrates the operational flow of the main module of **Fig. 1** for generating a drop down list of supported languages for changing the preferred language, in further detail, in accordance with one embodiment;

**Figures 10a-10b** illustrate various example data organizations suitable for use to store the message and pick list elements for practicing the present invention, in accordance with one embodiment;

**Figures 11a-11b** illustrate various example data organizations suitable for use to store the location information of the graphics/images and static pages for practicing the present invention, in accordance with one embodiment;

**Figure 12** illustrates an example computer system suitable for use to practice the present invention, in accordance with one embodiment; and

**Figure 13** illustrates an example network for practicing the present invention, in accordance with one embodiment.

#### DETAILED DESCRIPTION OF THE INVENTION

In the following description, various aspects of the present invention will be described. However, it will be apparent to those skilled in the art that the present invention may be practiced with only some or all aspects of the present invention. For purposes of explanation, specific numbers, materials and configurations are set forth in order to provide a thorough understanding of the present invention. However, it will also be apparent to one skilled in the art that the present invention may be practiced without the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

Parts of the description will be presented in terms of operations performed by a processor based device, using terms such as data, tables, requesting, determining, retrieving, displaying, and the like, consistent with the manner commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. As well understood by those skilled in the art, the quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, and otherwise manipulated through mechanical and electrical components of the processor based device; and the term processor include microprocessors, micro-controllers, digital signal processors, and the like, that are standalone, adjunct or embedded.

Various operations will be described as multiple discrete steps in turn, in a manner that is most helpful in understanding the present invention, however, the order of description should not be construed as to imply that these operations are necessarily order dependent. In particular, these operations need not be performed in the order of presentation. Further, the description repeatedly uses the phrase "in one embodiment", which ordinarily does not refer to the same embodiment, although it may.

### Overview

Referring now first to **Figure 1**, wherein a diagram illustrating an overview of the present invention, in accordance with one embodiment, is shown. As illustrated, content server **100** is advantageously equipped with one or more databases **112** having stored therein content elements, and content delivery engine **110** to deliver contents to client devices, e.g. client device **102**, based on a user's preferred language. In accordance with the present invention, content elements stored in databases **112** are expressed in and specifically designed for the supported languages. Content delivery engine **110** dynamically constructs the preferred language based contents to be delivered using the language specific content elements. More specifically, content elements stored in databases **112** include graphics/images specifically designed for the various supported languages, textual message and pick list elements as well as static content pages expressed in the supported



languages; and content delivery engine **110** includes a main module **120** incorporated with the teachings of the present invention, and supported by a number of support functions **122** including in particular, a Create\_Header function, a Get\_Message function, and a Get\_PickList Function to dynamically construct the required contents. As a result, content delivery engine **110** may deliver content based on a user's preferred language for a large number of languages. In one embodiment, content server **100** delivers dynamically constructed contents to client devices, based on a user's preferred language, supporting as many as 33 languages at the same time, a significantly higher number than any known prior art systems using prior art techniques. From the description to follow, those skilled in the art would appreciate that the present invention is highly scalable, and may easily be extended to support even more languages at the same time. In fact, there is no theoretical limit to the number of languages that can be currently supported by the present invention, as long as the necessary storage, computing resources, and communication bandwidth are available.

#### Preferred Language Based Content

**Figure 2** illustrates an example preferred language based content dynamically constructed in accordance with one embodiment of the present invention. As illustrated, if included, graphics/images **212** of preferred language based content **210** are dynamically constructed from graphics/images specifically designed for the various languages. That is, different graphics/images sensitive to the culture and community standards of the various communities speaking the supported languages are provided for the different languages. Further, if included, textual contents **214** of preferred language based content **210** are dynamically constructed from message elements expressed in the supported languages. That is, the contents are constructed from individual translated messages expressed in the supported languages.

Moreover, if included, links **216** are generated in the preferred language. That is secondary contents are accessed with resource location names

expressed in the supported languages. Similarly, if included, pick lists **218** are ordered in view of the preferred language, and static pages **224** of pop-ups **222** are expressed in the preferred language. That is, pick list elements are ordered depending on whether the preferred language is an alphabet based or a character based language, and the options are ordered in accordance with the manner the language customarily order items, e.g. alphabetically for English, by character strokes for Kanji (Japanese), and so forth.

Further, display space efficient drop-down list **220** (as opposed to the prior art display space inefficient button/icon approach) is advantageously employed to facilitate a user in changing the preferred language.

Finally, if included, pop-up static pages **224** are also generated in the preferred language. That is, different pop-up static pages expressed in the corresponding supported languages are employed for different preferred languages.

#### Operational Flow of Main

**Figure 3** illustrates the operational flow of the relevant aspects of the main module of **Fig. 1** for delivering content in a preferred language basis, in accordance with one embodiment of the present invention. As illustrated, at block **302** main **120** (using e.g. in-line instructions) sets one of the supported languages as the default language. In one embodiment, the default language is English. In other embodiments, the default language is the last language selected by the user as the preferred language. Various approaches known in the art may be employed to allow main **120** to remember the preferred language of a user across sessions.

At block **304**, main **120** sets the character code setting to the required character code set based on the preferred language. For the illustrated embodiment, main **120** sets the character code setting using the Create\_Header support function. More specifically, in an embodiment, where the contents are encoded using the Hypertext Markup Language (HTML), main **120**, using Create\_Header function, sets the character code setting by

generating a meta statement for the HTML header, specifying the required character code set.

Thereafter, at block **306**, if applicable, main **120** generates the graphics/images of the preferred language content, by retrieving appropriate ones of the stored graphics/images, which as described earlier, are specifically designed for the supported languages. For the illustrated embodiment, the different graphics/images are differentiated employing a graphic/image identification convention that includes the target language. Main **120** constructs the preferred language dependent graphic/image identifier to retrieve the appropriate ones of the graphics/images.

At block **308**, if applicable, main **120** generates the textual contents of the preferred language content, by retrieving appropriate ones of the stored message elements, which as described earlier, are expressed in the supported languages. For the illustrated embodiment, main **120** retrieves the appropriate ones of the message elements, using the `Get_Message` function.

At block **310**, if applicable, main **120** generates the links in the preferred language, by retrieving the appropriate language specific portion of the resource location names from databases **122**, and combining them with the non-language specific portions to form the preferred language tailored links. For the illustrated embodiment, main **120** also retrieves the appropriate ones of the language specific portions of the resource location names, using the `Get_Message` function.

At block **312**, if applicable, main **120** generates the pick lists, by retrieving the pick list elements from databases **122**, and ordering the retrieved pick list elements in accordance with the preferred language's ordering convention. For the illustrated embodiment, main **120** retrieves the appropriate ones of the pick list elements, and order them, using the `Get_List` function.

At block **314**, main **120** generates the drop down language selection list to facilitate a user in changing the preferred language.

Thereafter, main **120** awaits for user interactions with the content displayed, block **316**. Upon receipt of user inputs, main **120** determines if the inputs are associated with the user selecting one of the languages listed on the

drop down language selection list to change the preferred language, block **318**. If it is determined that the user is changing the preferred language, main **120** returns to block **304**, re-performs the earlier described operations illustrated by blocks **304-318**. If it is determined that the user is not changing the preferred language, the inputs are processed and handled in an application dependent manner, as in the prior art.

Accordingly, main **120**, augmented by support functions **122**, is able to dynamically generate contents based on a user's preferred language, supporting a large number of languages at the same time. These operations, and other related topics, will be described in turn in more detail below.

#### Setting the Character Code Setting

As appreciated by those skilled in the art, content expressed in different languages required different character code sets. For examples, to display contents in French, a character code set such as ISO-8859-1 is required, to display contents in Japanese, a character code set such as Shift-JIS is required, and so forth. As described earlier, in one embodiment, where the dynamically created contents are encoded using HTML, main **120** uses Create\_Header function to select the appropriate character code set for the preferred language.

**Figure 4** illustrates the operational flow of the relevant aspects of the Create\_Header function of **Fig. 1** for setting the character code setting for a preferred language, in accordance with an HTML embodiment of the present invention. As illustrated, at block **402**, upon invocation and provided with the current preferred language (\$language), the Create\_Header function, equipped with the necessary logic, sets the character code setting (\$charset) to the identifier of the appropriate character code set.

In one embodiment, the relevant logic of Create\_Header for setting \$charset may be as follows:

```
if ($language eq "E", "FR", "DE", "SP" etc.) {$charset = 'ISO-8859-1';}
    elseif ($language = 'JP') {$charset = 'Shift-JIS';}
    elseif ($language = 'CHT') {$charset = 'Big5';}
    etc.
```

else {\$charset = 'UTF-8'}

At block **404**, upon setting \$charset, the Create\_Header function proceeds to generate the HTML header, which is known in the art. However, in accordance with the present invention, at block **406**, the Create\_Header function further appends a meta statement to the end of the HTML header being generated, specifying the required character code set.

Accordingly, main **120** may dynamically set the character code setting for a large number of languages.

#### Generating Graphics/Images for a Preferred Language

As described earlier, the present invention advantageously generates graphics/images that are specifically designed for the various languages, taking into consideration the culture and community standards of the communities speaking the supported languages. For the illustrated embodiment, the language specific graphics/images employ a naming convention that includes the target language. Accordingly, main **120** dynamic generates the appropriate graphics/images by dynamically generating the appropriate language specific references to the required graphics/images.

**Figure 5** illustrates the operational flow of main **120** for generating a preferred language based graphic/image, in accordance with one embodiment of the present invention. At block **502**, main **120** performs the initial setup, identifying a page location for a graphic/image to be included. At block **504**, main **120** generates the preferred language based identifier identifying the appropriate version of the desired graphics/images. Main **120** assigns the generated identifier to a filename variable (\$the\_file). The generation and assignment may e.g., be accomplished using the indirect assignment statement:

```
"$the_file = $logo_path . '/logo_' . lc($lang) . '.gif',
```

where the function "lc" returns a language code, such as "FR" for French, "JP" for Japanese, and so forth, depending on whether the preferred language (\$lang) is "French" or "Japanese", and so forth.

At block **506**, upon setting \$this\_file, main **120** retrieves and includes the retrieved preferred language specific graphics/images in the content being

dynamically generated. The inclusion may e.g., be accomplished using the instruction:

```
print "<img src = '$the_file' alt='alt_filename' etc">",
```

where "alt" specifies an alternate graphic/image, in the event there is no specifically designed graphic/image for the desired graphic/image.

Accordingly, main **120** may dynamically generate language specific graphics/images for a large number of languages.

#### Generating Textual Contents for a Preferred Language

As described earlier, the present invention advantageously generates the textual contents using message elements expressed in the preferred language. More specifically, main **120** repeatedly invokes the Get\_Message function to retrieve the required message elements (expressed in the preferred language) to construct the required textual contents.

**Figure 6** illustrates the operational flow of the Get\_Message function for returning a message or message element desired by main **120**, in accordance with one embodiment of the present invention. For the illustrated embodiment, it is assumed that each message element is associated with a message identifier, identifying the message element.

At block **602**, upon invocation and provided with the identifier (msg#) of the message element to be retrieved, and the preferred language (\$language) the message is to be expressed in, the Get\_Message function performs the initial setup, setting the current message identifier (\$this\_message\_id) for the current message element to be retrieved to the provided message identifier (msg#), and the current language (\$this\_message\_language) to the preferred language (\$language) provided.

At block **604**, the Get\_Message function queries the message table of database **112**, and retrieves the specified message and its associated message attributes, such as the message's font, alignment, and so forth.

The retrieval may be accomplished using the following instruction:

```
"SELECT message_text, message attributes, etc.  
FROM Message_Table
```

WHERE message\_id = \$this\_message\_id and message\_language  
= '\$this\_message\_lang'.

For the illustrated embodiment, it is assumed that each message may be returned as one or more message segments. At blocks **606-610**, the Get\_Message function successively accepts the various segments of the requested message returned from database **112**, using a returned message array. Thereafter, having received all the segments of the requested message, the Get\_Message function returns the retrieved message to main **120** (not shown).

Accordingly, using Get\_Message function, main **120** may dynamically generate textual contents expressed in a large number of languages.

#### Generating Links in a Preferred Language

As described earlier, the present invention advantageously generates the included links to additional contents in the user's preferred language. **Figure 7** illustrates the operational flow of main **120** for generating a link in the preferred language, in accordance with one embodiment of the present invention. For the illustrated embodiment, it is assumed that the additional content employs a resource location identification convention that at least partially expresses the identification in the language of the content. That is, if the additional content is expressed in French, then the resource location identifier of the additional content is also at least partially expressed in French. Correspondingly, if the additional content is expressed in Japanese, then the resource location identifier of the additional content is also at least partially expressed in Japanese.

At block **702**, main **120** retrieves the language specific portion (\$lang\_specific\_portion) of the link to be generated using the earlier described Get\_Message function. That is, for the illustrated embodiment, the language specific portions of the links to additional contents (expressed in various languages) are also stored in the message table of database **112**, as other message elements employed to construct the textual contents.

At blocks **704-706**, upon receipt of the language specific portion of the link to be generated, main **120** generates the preferred language based link by

combining the retrieved language specific portion of the link with non-language specific portion of the link, and outputting the combined link reference. The combination and generation, may e.g., be accomplished through the following assignment and print instructions:

```
"$the_link = <a href = 'non-language specific portion text' . lc($language) .  
$lang_specific_portion /a>";  
print "<formatting options >$the_link< more formatting options>".
```

Accordingly, using Get\_Message function, main **120** may dynamically generate links in the preferred language for a large number of languages.

#### Generating Pick List in view of a Preferred Language

As described earlier, the present invention advantageously generates pick lists in view of a user's preferred language. More specifically, main **120** generates and ordered a desired pick list in view of the preferred language, using Pick\_List function of **Fig. 1**.

**Figure 8** illustrates the operational flow of the relevant aspects of the Pick\_List function for generating a pick list in view of in the preferred language, in accordance with one embodiment of the present invention. For the illustrated embodiment, it is assumed that at least a list sequence attribute is associated with each of the pick list elements of non-alphabet languages, such as Japanese, Chinese, and so forth.

At block **802**, upon invocation and provided with the current preferred language, the Get\_List function sets the current pick list language (\$list\_lang) to the provided preferred language. Thereafter, at block **804**, the Get\_List function sets an order criterion based on the preferred language. More specifically, in one embodiment, the order criterion is set to "order by sequence" if the preferred language is a non-alpha language, otherwise the order criterion is set to "order by text". At block **806**, the Get\_List function retrieves the appropriate pick list elements from the pick list table of database **112**, and order the pick list be retrieved in accordance with the order criterion set. The retrieval and ordering may be accomplished e.g. by the following instruction:

```
SELECT list_text, etc.
```



FROM List\_Table

WHERE list\_id=\$this\_list\_id and list\_language='\$this\_list\_lang', \$order.

For the illustrated embodiment, a list element array is employed to receive the returned pick list elements. At blocks **808-810**, the Get\_List function successively receives the returned pick list elements.

Accordingly, using Get\_List function, main **120** may dynamically generate pick lists ordered in accordance with the preferred language. The manner in which the dynamic generation is accomplished, allows main **120** to do so for a large number of languages at the same time.

#### Generating A Drop Down List of Supported Languages for Changing The Preferred Language

As described earlier, the present invention advantageously generates a space saving drop down list of supported languages to facilitate a user in changing the preferred language. For the illustrated embodiment, main **120** generates the drop down list of languages and capture a user's desire to change the preferred language using a form approach. Upon selection of one of the displayed language, the form is submitted to main **120**, which in response causes new contents to be dynamically re-generated as earlier described based on the new preferred language. In one embodiment, main **120** employs a "Redirect" Control Gateway Interface (CGI) script to trigger the dynamic regeneration.

The form based approach may be accomplished e.g. using the following instructions:

```
<form method="POST" action=protocol://sitename/cgi-bin/redirect.cgi target="_parent">
  <div align="center"><center><p><select
name="lang_pref" size="1" onChange="form.submit()">
  <option value="af">Afrikaans</option>
  <option value="ar">Arabic</option>
  <option value="cz">Cesky</option>
  <option value="chm">Chinese (Modern)</option>
```

```

<option value="cht">Chinese (Traditional)</option>
<option value="cr">Croatian</option>
<option value="da">Dansk</option>
<option value="de">Deutsch</option>
<option selected value="e">English</option>
<option value="sp">Español</option>
<option value="fr">Français</option>
<option value="el">Hellas</option>
<option value="hi">Hindi</option>
<option value="nd">Indonesian</option>
<option value="it">Italiano</option>
<option value="jp">Japanese</option>
<option value="kr">Korean</option>
<option value="hu">Magyar</option>
<option value="nl">Nederlands</option>
    etc.
    <option value="vt">Vietnamese</option>
</select></p>
</center></div>
</form>

```

**Figure 9** illustrates the operational flow of the relevant aspects of the Redirect CGI script. As illustrated, upon invocation and provided with the newly selected preferred language, at block **902**, the Redirect CGI script sets the current preferred language (\$lang) with the provided language (lang\_pref). Thereafter, at block **904**, the Redirect CGI script sets a path variable (\$new\_path) accordingly, based on the current preferred language, pointing to the appropriate new content to be dynamically generated. The path variable may be set, e.g. using the following conditional assignment instruction:

```

if ($language eq "E", "FR", "DE", "SP" etc.)
    {$new_path = 'http://first_portion/' . $lang . 'html';}
elseif ($language = 'JP') {$new_path = 'pathname';}
elseif ($language = 'CHT') {$new_path = 'pathname';}

```

etc.

Upon setting the new path for the new content, Redirect CGI script causes the new content to be dynamically generated by invoking the new content. The invocation may e.g. be accomplished using the following instruction:

```
print "Location: $new_path\n\n".
```

Accordingly, main **120** is able to facilitate a user in changing the preferred language in a space saving, compact and efficient manner, therefore, allowing main **120** to facilitate a user in changing the preferred language to one of a large number of supported languages.

### Generating Pop-up Static Pages

As described earlier, the present invention advantageously generates pop-up static pages expressed in the user's preferred language. For the illustrated embodiment, similar to the language specific graphics/images, pop-up static pages also employ a naming convention that includes the target language. Accordingly, main **120** dynamic generates the pop-up static pages expressed in the preferred language, the same manner it dynamically generates the appropriate graphics/images, by dynamically generating the appropriate language specific references to the required pop-up static pages.

### Sample Data Organizations

**Figures 10a-10b** illustrate two example data organizations suitable for use to store the message elements and the pick list elements for practicing the present invention. As illustrated in **Fig. 10a**, message table **1000** includes column **1002** and column **1004** for storing the message identifiers and the language identifiers of the stored messages or message elements. Message table **1000** further includes column **1006** for storing the message texts of the messages, and columns **1008** for storing the message attributes, such as fonts, alignments, and so forth associated with the stored messages.

Similarly, as illustrated in **Fig. 10b**, pick list table **1010** includes column **1012** and column **1014** for storing the pick list identifiers and the language

identifiers of the stored pick list elements. Pick list table **1010** further includes column **1016** for storing the texts of the pick list elements, column **1018** for storing list sequence values for the pick list elements (in particular, for the pick list elements of non-alphabet languages), and columns **1020** for other related data.

In one embodiment, tables **1000** and **1010** are relational tables of databases **112**. In alternate embodiments, other equivalent data structures may be used instead.

**Figures 11a-11b** illustrate two example directory organizations suitable for use to store the location information pointing to the language specific graphic/image files and static content pages. As illustrated in **Fig. 11a**, directory **1100** includes field **1102** and field **1104** for storing the file identifiers and file attributes of the language specific graphic/image files **1110**. As described earlier, the file identifiers employ a naming convention that includes identification information of the target language. Examples of file attributes include creation date, file size, and so forth. Directory **1100** also includes field **1106** for storing the location of the graphic/image files **1110**, e.g. the cylinder and tack location of a disk drive.

As shown in **Fig. 11b**, directory **1120** includes field **1122** and field **1124** for storing the file identifiers and file attributes of the static content pages expressed in the various supported languages. As described earlier, the file identifiers employ a naming convention that includes identification information of the target language. Similarly, examples of file attributes include creation date, file size, and so forth. Directory **1120** also includes field **1126** for storing the location of the static pages, e.g. the cylinder and tack location of a disk drive.

#### Example Computer System

**Figure 12** illustrates an example computer system suitable for use to practice the present invention, in accordance with one embodiment. As shown, computer system **1200** includes one or more processors **1202** and system memory **1204**. Additionally, computer system **1200** includes mass storage

devices **1206** (such as diskette, hard drive, CDROM and so forth), input/output devices **1208** (such as keyboard, cursor control and so forth) and communication interfaces **1210** (such as network interface cards, modems and so forth). The elements are coupled to each other via system bus **1212**, which represents one or more buses. In the case of multiple buses, they are bridged by one or more bus bridges (not shown). Each of these elements performs its conventional functions known in the art. In particular, system memory **1204** and mass storage **1206** are employed to store a working copy and a permanent copy of the programming instructions implementing the content delivery engine of the present invention. The permanent copy of the programming instructions may be loaded into mass storage **1206** in the factory, or in the field, as described earlier, through a distribution medium (not shown) or through communication interface **1210** (from a distribution server (not shown)). The constitution of these elements **1202-1212** are known, and accordingly will not be further described.

#### Example Network Environment

**Figure 13** illustrates an example network environment suitable for use to practice the present invention, in accordance with one embodiment. As illustrated, network environment **1300** includes content server **1302** and client devices **1304a-1304b** coupled to each other via internetworking fabric **1306**. Content server **1302** equipped with the content delivery engine of the present invention, advantageously delivers contents to client devices **1304a-1304b** in accordance with the preferred languages of client devices **1304a-1304b**, as described earlier.

Except for the content delivery engine of the present invention, content server **1302** is intended to represent a broad range of server computers known in the art, including example computer system illustrated in **Fig. 12**. Similarly, client devices **1304a-1304b** are intended to represent a broad range of palm sized, notebook sized, and desktop computing devices, as well as set-top boxes. In one embodiment, internetworking fabric **1306** represents the

Internet, whereas in alternate embodiments, internetworking fabric **1306** may be a private network, e.g. an enterprise network.

In various embodiments, networking environment **1300** including content server **1302** and client devices **1304a-1304b** are employed to create, conduct, collect, analyze and report on consumer/marketing research/surveys in a large number of languages. In particular, in one embodiment, network environment **1300** is employed to create surveys as described in co-pending U.S. Patent Application, <number to be assigned>, entitled Multi-Region Market Research Study Creation filed contemporaneously, and having common inventorship with the present invention. In another embodiment, network environment **1300** is employed to conduct surveys as described in co-pending U.S. Patent Application, <number to be assigned>, entitled Multi-Region Market Research Study Processing, filed contemporaneously, and having common inventorship with the present invention. In yet another embodiment, network environment **1300** is employed to analyze and report on survey results as described in co-pending U.S. Patent Application, <number to be assigned>, entitled Reporting And Analyzing Data From A Multi-Region Research Survey, filed contemporaneously, and having common inventorship with the present invention. These co-pending applications are all hereby fully incorporated by reference.

#### Conclusion and Epilogue

Thus, it can be seen from the above descriptions, a novel method and apparatus for delivering content in a preferred language has been described. The novel method/apparatus is advantageously scalable to support a large number of languages.

While the present invention has been described in terms of the above illustrated embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described. The present invention can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of restrictive on the present invention.

## CLAIMS

What is claimed is:

1. A method for real time content delivery in a preferred language chosen from a large number of supported languages, the method comprising:
  - setting a language as a preferred language;
  - setting a character code setting of a content page based on the preferred language;
  - generating graphic/image contents of the content page by accessing a graphic/image library having a plurality of language specific graphics/images specifically designed for the supported languages, and retrieving one or more of the language specific graphics/images from the graphic/image library based on the preferred language;
  - generating textual contents of the content page by accessing a message library having a plurality of messages expressed in the supported languages, and retrieving one or more of the messages from the message library based on the preferred language;
  - generating one or more links expressed in the preferred language for the content page, forming said links with non-language specific portions and language specific portions of said links retrieved based on the preferred language; and
  - generating a drop down list of languages for the content page for changing the preferred language to another one of said supported languages.
2. The method of claim 1, wherein said content page is encoded using a markup language that includes a header element and a meta statement for the header element, and setting of the character code setting comprises determining a character code set based on the preferred language, and generating the header element including a meta statement setting a character code setting to the determined character code set.

3. The method of claim 1, wherein said accessing and retrieving of graphics/images comprises generating for each graphic/image to be retrieved, a language code based on the preferred language, forming a graphic/image filename for the graphic/image to be retrieved, including the language code in a predetermined location of the graphic/image filename being formed, and retrieving a graphic/image specifically designed for the preferred language using the graphic/image filename formed.
4. The method of claim 1, wherein accessing and retrieving of messages comprises retrieving a plurality of messages expressed in the preferred language from the message library, with each message expressed in the preferred language being retrieved from the message library using a message identifier and an identifier of the preferred language.
5. The method of claim 4, wherein retrieval of each message expressed in the preferred language comprises retrieving a plurality of message segments of the message expressed in the preferred language along with a plurality of message attributes, and formatting each of the message segments in accordance with the message attributes.
6. The method of claim 1, where said retrieving of language specific portions of the links comprises retrieving each of the language specific portion of a link from a message table using a message identifier and an identifier of the preferred language.
7. The method of claim 1, wherein the method further comprises re-performing said setting and generating operations, whenever one of the listed languages of the drop down list is selected.
8. The method of claim 1, wherein the method further comprises generating one or more pick lists for the content page, with each of the one or more pick lists being organized in view of the preferred language.



9. The method of claim 8, wherein said generation of each of the pick lists comprises setting an order criterion based on the preferred language, and retrieving list elements of the pick list from a pick list element table in accordance with the order specified by the order criterion.

10. A method for real time content delivery in a preferred language chosen from a large number of supported languages, the method comprising:

setting a language as a preferred language;

determining a character code set for a content page based on the preferred language; and

generating a meta statement of a header element of the content page setting a character code setting to the determined character code set.

11. The method of claim 10, wherein the method further comprises generating graphic/image contents of the content page by accessing a graphic/image library having a plurality of language specific graphics/images specifically designed for the supported languages, and retrieving one or more of the language specific graphics/images from the graphic/image library based on the preferred language.

12. The method of claim 10, wherein the method further comprises generating textual contents of the content page by accessing a message library having a plurality of messages expressed in the supported languages, and retrieving one or more of the messages from the message library based on the preferred language.

13. The method of claim 10, wherein the method further comprises generating one or more links expressed in the preferred language for the content page.

14. The method of claim 10, wherein the method further comprises generating one or more pick lists of the content page with each of the pick lists being ordered in view of the preferred language.
15. The method of claim 10, wherein the method further comprises generating a drop down list of languages for the content page for changing the preferred language to another one of the supported languages.
16. A method for real time content delivery in a preferred language chosen from a large number of supported languages, the method comprising:
  - setting a language as a preferred language; and
  - generating graphic/image contents of a content page by accessing a graphic/image library having a plurality of language specific graphics/images specifically designed for the supported languages, and retrieving one or more of the language specific graphics/images from the graphic/image library based on the preferred language.
17. The method of claim 16, wherein the method further comprises setting a character code setting for the content page based on the preferred language.
18. The method of claim 16, wherein the method further comprises generating textual contents of the content page by accessing a message library having a plurality of messages expressed in the supported languages, and retrieving one or more of the messages from the message library based on the preferred language.
19. The method of claim 16, wherein the method further comprises generating one or more links expressed in the preferred language for the content page.

20. The method of claim 16, wherein the method further comprises generating one or more pick lists of the content page with each of the pick lists being ordered in view of the preferred language.

21. The method of claim 16, wherein the method further comprises generating a drop down list of languages for the content page for changing the preferred language to another one of the supported languages.

22. A method for real time content delivery in a preferred language chosen from a large number of supported languages, the method comprising:  
    setting a language as a preferred language; and  
    generating textual contents of the content page by accessing a message library having a plurality of messages expressed in the supported languages, and retrieving one or more of the messages from the message library based on the preferred language.

23. The method of claim 22, wherein the method further comprises setting a character code setting for the content page based on the preferred language.

24. The method of claim 22, wherein the method further comprises generating graphic/image contents of a content page by accessing a graphic/image library having a plurality of language specific graphics/images specifically designed for the supported languages, and retrieving one or more of the language specific graphics/images from the graphic/image library based on the preferred language.

25. The method of claim 22, wherein the method further comprises generating one or more links expressed in the preferred language for the content page.

26. The method of claim 22, wherein the method further comprises generating one or more pick lists of the content page with each of the pick lists being ordered in view of the preferred language.

27. The method of claim 22, wherein the method further comprises generating a drop down list of languages for the content page for changing the preferred language to another one of the supported languages.

28. A method for real time content delivery in a preferred language chosen from a large number of supported languages, the method comprising:  
    setting a language as a preferred language;  
    retrieving a language specific portion for each of one or more links to be generated for a content page from a message table using a message identifier and an identifier of the preferred language; and  
    generating the one or more links for the content page, for each of the one or more links, combining a non-language specific portion with the retrieved language specific portion to form the link.

29. The method of claim 28, wherein the method further comprises setting a character code setting for the content page based on the preferred language.

30. The method of claim 28, wherein the method further comprises generating graphic/image contents of a content page by accessing a graphic/image library having a plurality of language specific graphics/images specifically designed for the supported languages, and retrieving one or more of the language specific graphics/images from the graphic/image library based on the preferred language.

31. The method of claim 28, wherein the method further comprises generating textual contents of the content page by accessing a message library having a plurality of messages expressed in the supported languages,

and retrieving one or more of the messages from the message library based on the preferred language.

32. The method of claim 28, wherein the method further comprises generating one or more pick lists of the content page with each of the pick lists being ordered in view of the preferred language.

33. The method of claim 28, wherein the method further comprises generating a drop down list of languages for the content page for changing the preferred language to another one of the supported languages.

34. A method for real time content delivery in a preferred language chosen from a large number of supported languages, the method comprising:  
    setting a language as a preferred language;  
    determining an order criterion in view of the preferred language; and  
    retrieving a plurality of pick list elements from a pick list element table, ordering the retrieved pick list elements in accordance with the order criterion, to generate one or more pick lists for a content page.

35. The method of claim 34, wherein the method further comprises setting a character code setting for the content page based on the preferred language.

36. The method of claim 34, wherein the method further comprises generating graphic/image contents of a content page by accessing a graphic/image library having a plurality of language specific graphics/images specifically designed for the supported languages, and retrieving one or more of the language specific graphics/images from the graphic/image library based on the preferred language.

37. The method of claim 34, wherein the method further comprises generating textual contents of the content page by accessing a message library having a plurality of messages expressed in the supported languages,

and retrieving one or more of the messages from the message library based on the preferred language.

38. The method of claim 34, wherein the method further comprises generating one or more links expressed in the preferred language for the content page.

39. The method of claim 34, wherein the method further comprises generating a drop down list of languages for the content page for changing the preferred language to another one of the supported languages.

40. A method for real time content delivery in a preferred language chosen from a large number of supported languages, the method comprising:

setting a language as a preferred language;

generating contents of a content page based at least in part on the preferred language;

generating a drop down list of languages for the content page for changing the preferred language to another one of the supported languages;  
and

re-performing said generating of contents of the content page based at least in part of the preferred language, whenever one of the drop down list of languages is selected.

41. The method of claim 40, wherein the method further comprises setting a character code setting for the content page based on the preferred language.

42. The method of claim 40, wherein the method further comprises generating graphic/image contents of a content page by accessing a graphic/image library having a plurality of language specific graphics/images specifically designed for the supported languages, and retrieving one or more of the language specific graphics/images from the graphic/image library based on the preferred language.

43. The method of claim 40, wherein the method further comprises generating textual contents of the content page by accessing a message library having a plurality of messages expressed in the supported languages, and retrieving one or more of the messages from the message library based on the preferred language.

44. The method of claim 40, wherein the method further comprises generating one or more links expressed in the preferred language for the content page.

45. The method of claim 40, wherein the method further comprises generating one or more pick lists of the content page with each of the pick lists being ordered in view of the preferred language.

46. An apparatus comprising:

storage medium having stored therein a plurality of programming instructions designed to implement a plurality of support functions in support of a content delivery engine in delivering content in a preferred language, including

a first function to set a language as a preferred language,

a second function to set a character code setting of a content page based on the preferred language,

a third function to generate graphic/image contents of the content page by accessing a graphic/image library having a plurality of language specific graphics/images specifically designed for the supported languages, and retrieving one or more of the language specific graphics/images from the graphic/image library based on the preferred language,

a fourth function to generate textual contents of the content page by accessing a message library having a plurality of messages expressed in the supported languages, and retrieving one or more of the messages from the message library based on the preferred language,

a fifth function to generate one or more links expressed in the preferred language for the content page, forming said links with non-language specific portions and language specific portions of said links retrieved based on the preferred language, and

a sixth function to generate a drop down list of languages for the content page for changing the preferred language to another one of a plurality of supported languages; and

one or more processors coupled to the storage medium to execute the programming instructions.

47. The apparatus of claim 46, wherein said content page is encoded using a markup language that includes a header element and a meta statement for the header element, and the second function determines a character code set based on the preferred language, and generates the header element including a meta statement setting a character code setting to the determined character code set.

48. The apparatus of claim 46, wherein the third function generates for each graphic/image to be retrieved, a language code based on the preferred language, forms a graphic/image filename for the graphic/image to be retrieved, including the language code in a predetermined location of the graphic/image filename being formed, and retrieves a graphic/image specifically designed for the preferred language using the graphic/image filename formed.

49. The apparatus of claim 46, wherein the fourth function retrieves a plurality of messages expressed in the preferred language from the message library, with each message expressed in the preferred language being retrieved from the message library using a message identifier and an identifier of the preferred language.



50. The apparatus of claim 49, wherein the fourth function retrieves a plurality of message segments of the message expressed in the preferred language along with a plurality of message attributes, and formats each of the message segments in accordance with the message attributes.

51. The apparatus of claim 46, wherein the fifth function retrieves each of the language specific portion of a link from a message table using a message identifier and an identifier of the preferred language.

52. The apparatus of claim 46, wherein the first through sixth functions are re-executed, whenever one of the listed languages of the drop down list is selected.

53. The apparatus of claim 46, wherein the programming instructions further implement a seventh function to generate one or more pick lists for the content page, with each of the one or more pick lists being organized in view of the preferred language.

54. The apparatus of claim 53, wherein said seventh function sets an order criterion based on the preferred language, and retrieves list elements of the pick list from a pick list element table in accordance with the order specified by the order criterion.

55. An apparatus comprising:

storage medium having stored therein a plurality of programming instructions designed to implement a plurality of support functions in support of a content delivery engine in delivering content in a preferred language, including a first function to set a language as a preferred language, and a second function to determine a character code set based on the preferred language, and generate meta statement of a header element of a content page setting a character code setting to the character code set; and

one or more processors coupled to the storage medium to execute the programming instructions.

56. An apparatus comprising:

storage medium having stored therein a plurality of programming instructions designed to implement a plurality of support functions in support of a content delivery engine in delivering content in a preferred language, including a first function to set a language as a preferred language, and a second function to generate graphic/image contents of the content page by accessing a graphic/image library having a plurality of language specific graphics/images specifically designed for the supported languages, and retrieving one or more of the language specific graphics/images from the graphic/image library based on the preferred language; and

one or more processors coupled to the storage medium to execute the programming instructions.

57. An apparatus comprising:

storage medium having stored therein a plurality of programming instructions designed to implement a plurality of support functions in support of a content delivery engine in delivering content in a preferred language, including a first function to set a language as a preferred language, and a second function to generate textual contents of the content page by accessing a message library having a plurality of messages expressed in the supported languages, and retrieving one or more of the messages from the message library based on the preferred language; and

one or more processors coupled to the storage medium to execute the programming instructions.

58. An apparatus comprising:

storage medium having stored therein a plurality of programming instructions designed to implement a plurality of support functions in support of a content delivery engine in delivering content in a preferred language,

including a first function to set a language as a preferred language, and a second function to retrieve a language specific portion for each of one or more links to be generated for a content page, based on the preferred language, and combining one or more non-language specific portions with said retrieved one or more language specific portions to generate said one or more links for the content page; and

one or more processors coupled to the storage medium to execute the programming instructions.

59. An apparatus comprising:

storage medium having stored therein a plurality of programming instructions designed to implement a plurality of support functions in support of a content delivery engine in delivering content in a preferred language, including a first function to set a language as a preferred language, and a second function to determine an order criterion based on the preferred language, and retrieve a plurality of pick list elements from a pick list table, ordering the retrieved pick list elements in accordance with the determined order criterion to generate one or more pick lists for a content page; and

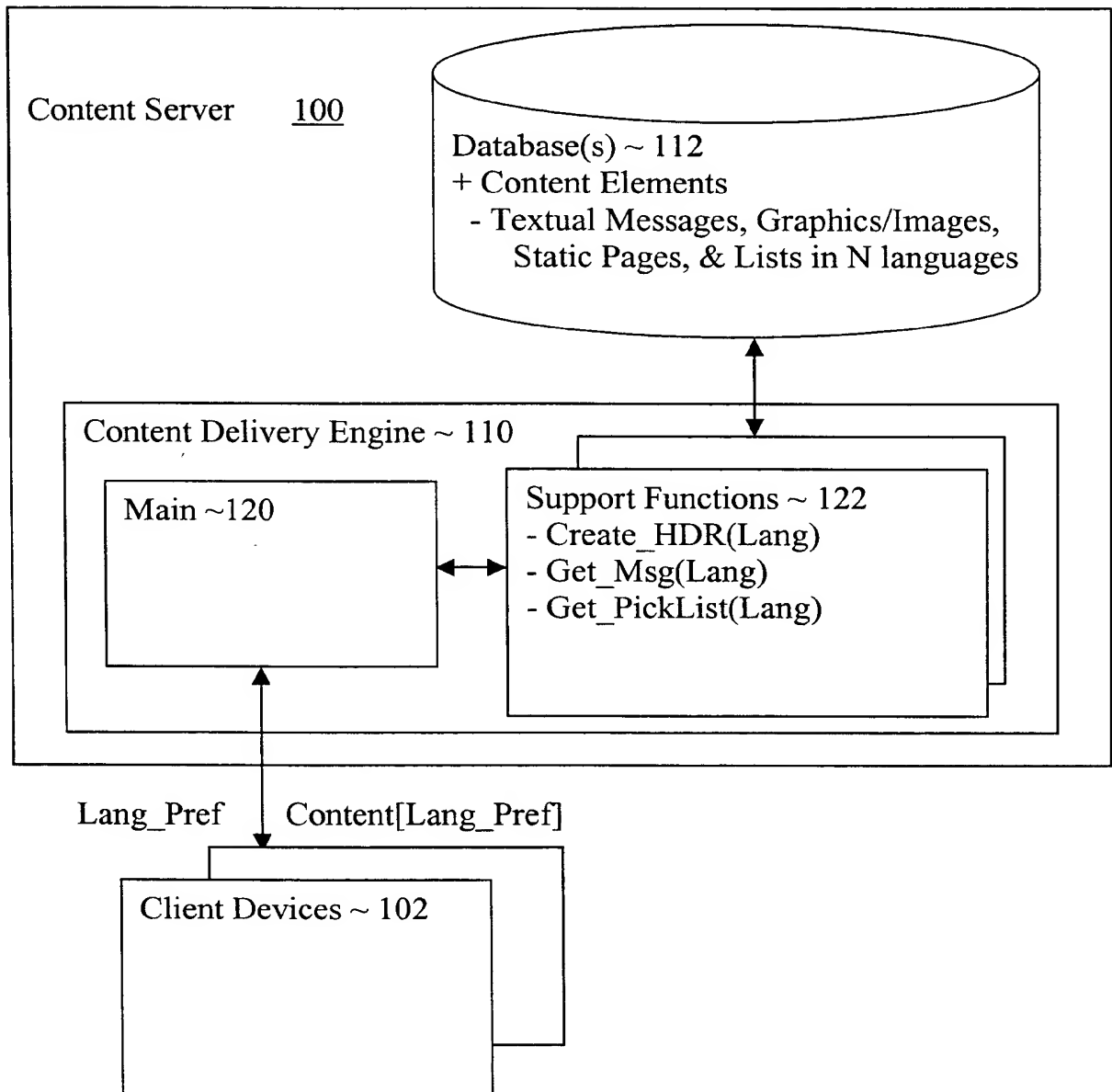
one or more processors coupled to the storage medium to execute the programming instructions.

60. An apparatus comprising:

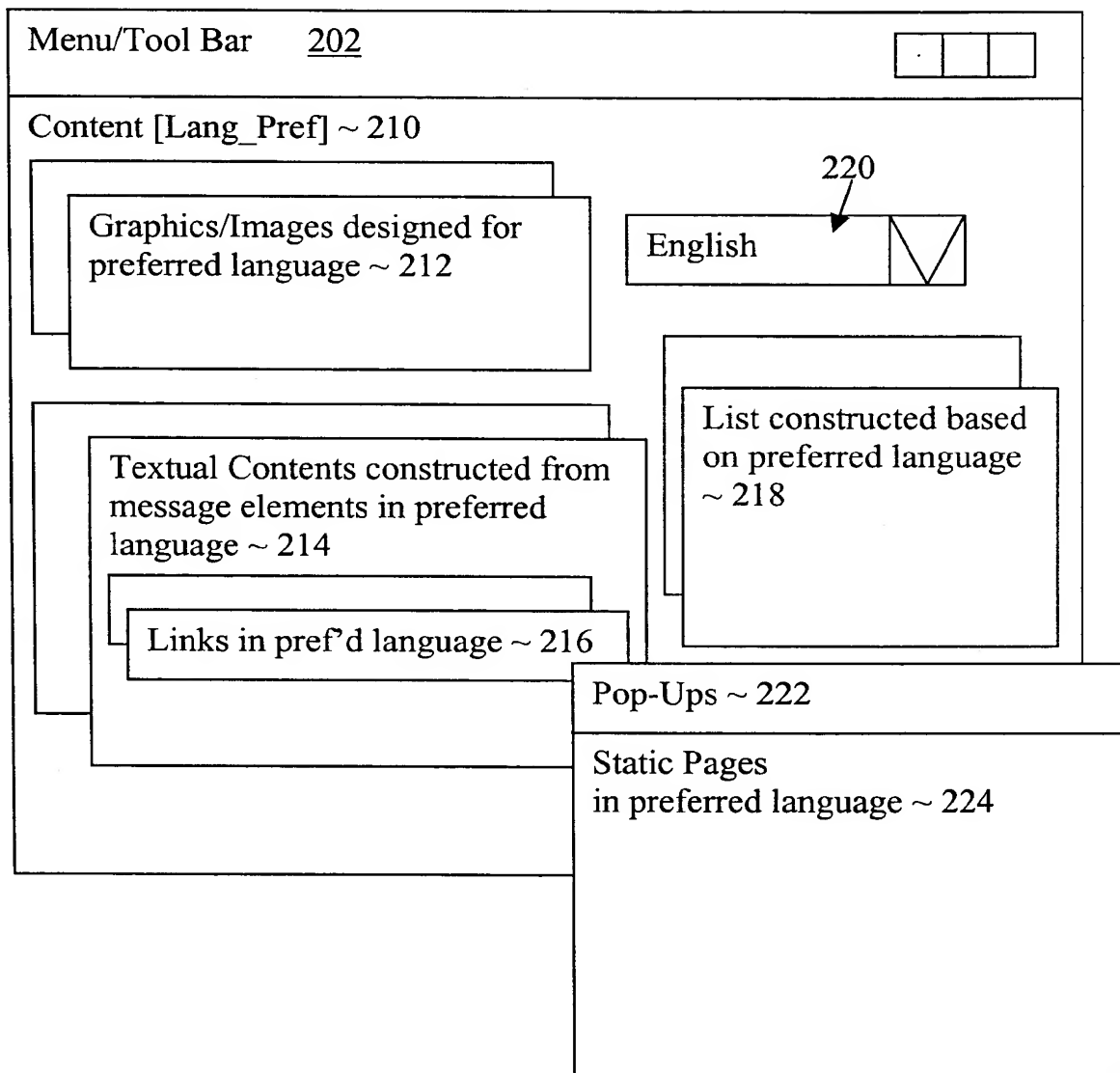
storage medium having stored therein a plurality of programming instructions designed to implement a plurality of support functions in support of a content delivery engine in delivering content in a preferred language, including a first function to set a language as a preferred language, a second function to generate a content page based on the preferred language, including a drop down list of languages for changing the preferred language to another one of a plurality of supported languages, and the second function regenerating the content page based on the preferred language whenever one of the drop down list of languages is selected; and

one or more processors coupled to the storage medium to execute the programming instructions.

1/13

**Figure 1**

2/13

200**Figure 2**

3/13

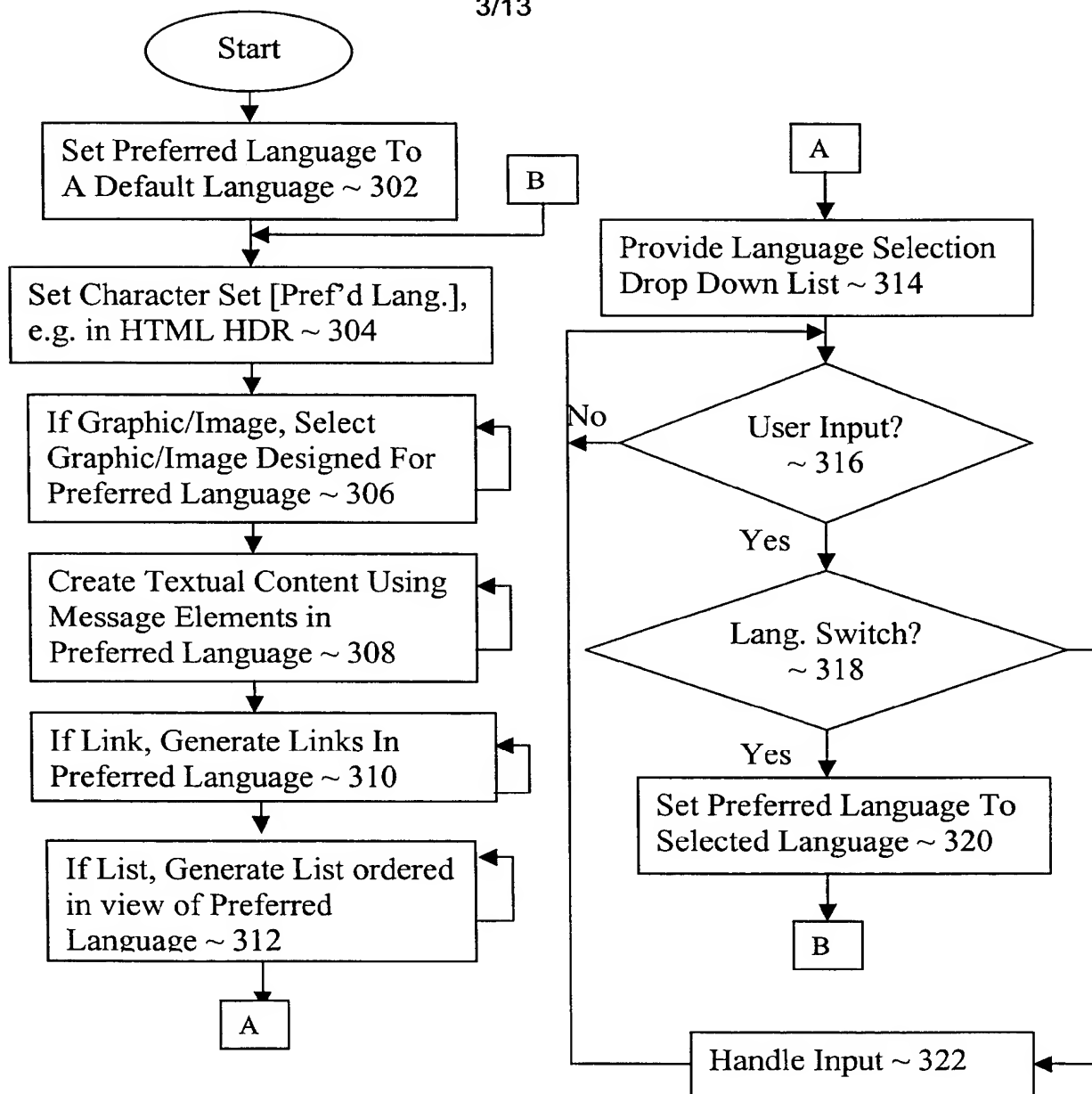
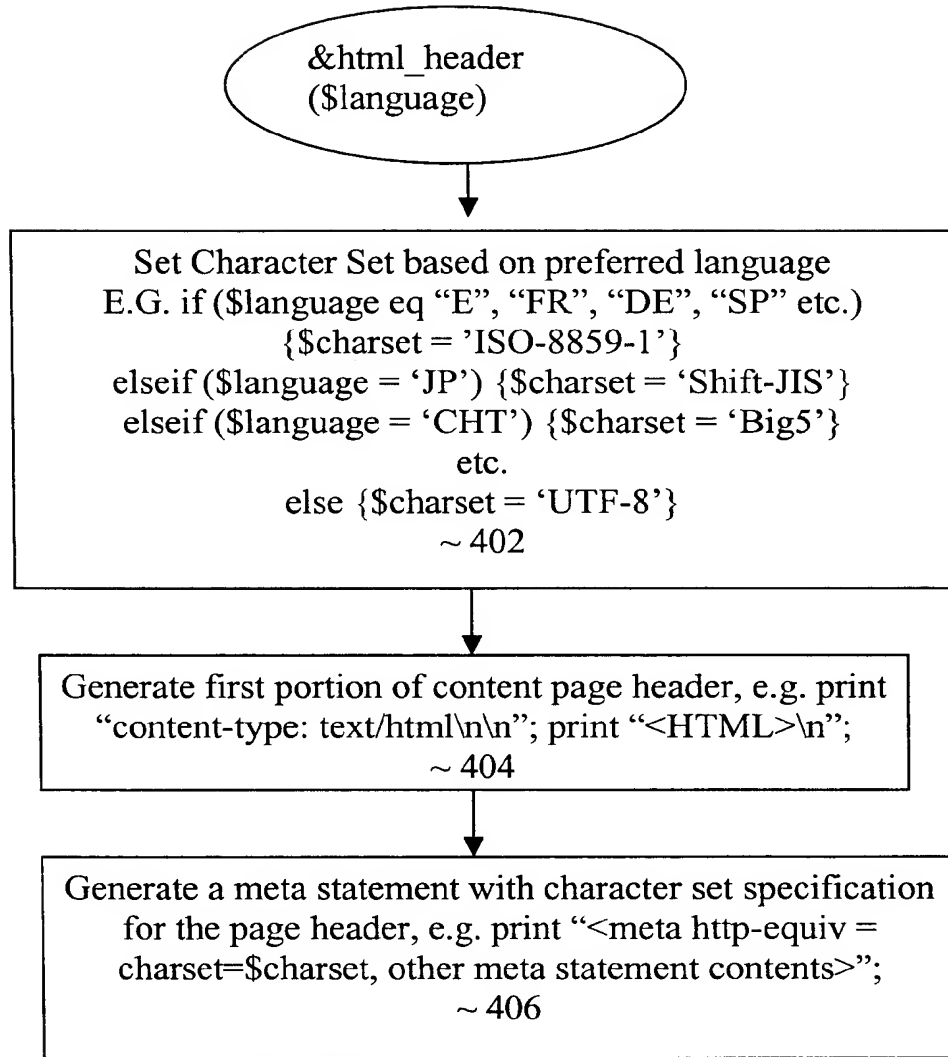


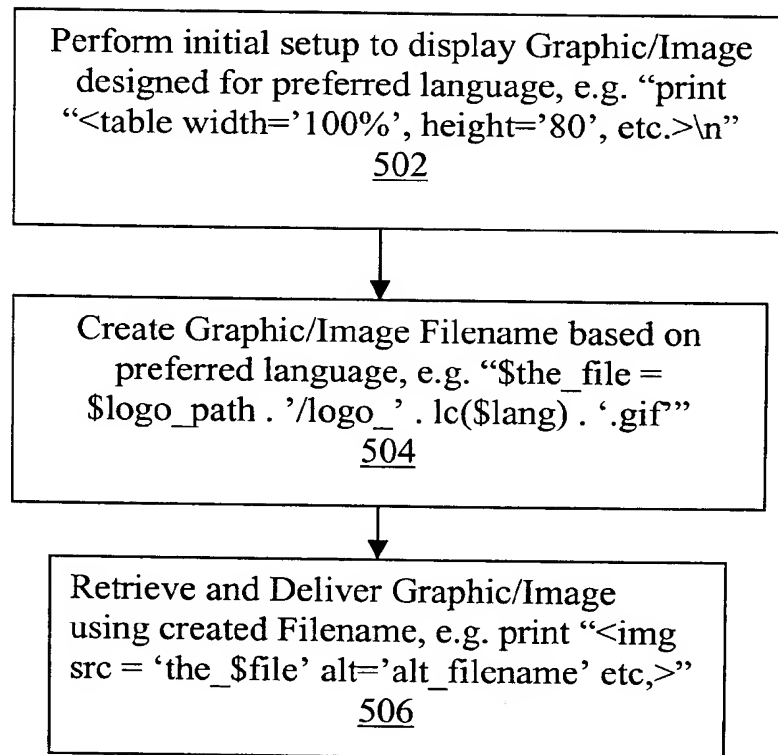
Figure 3

4/13

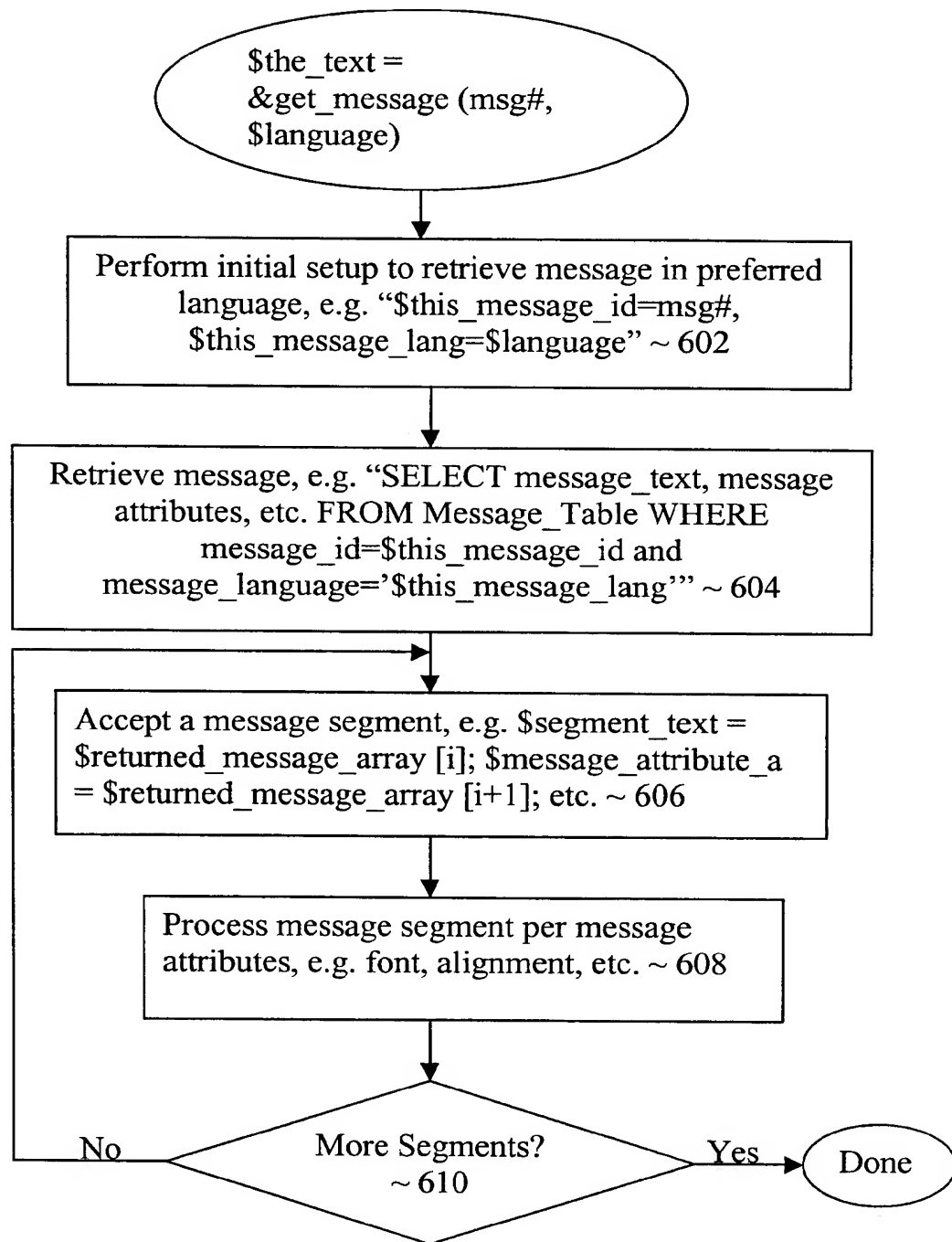
**Figure 4**



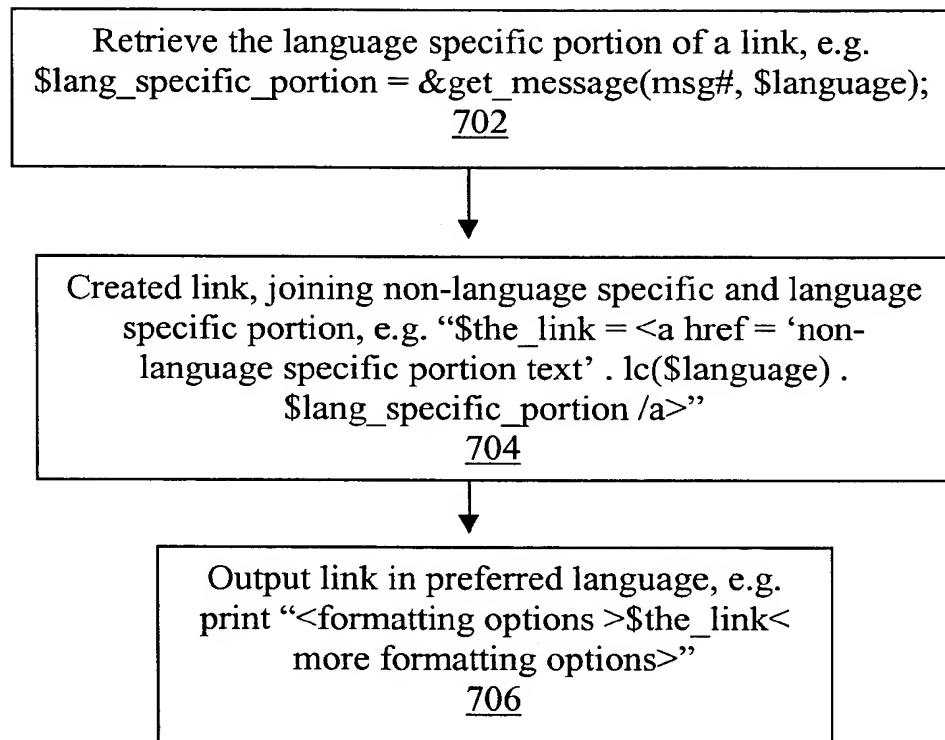
5/13

**Figure 5**

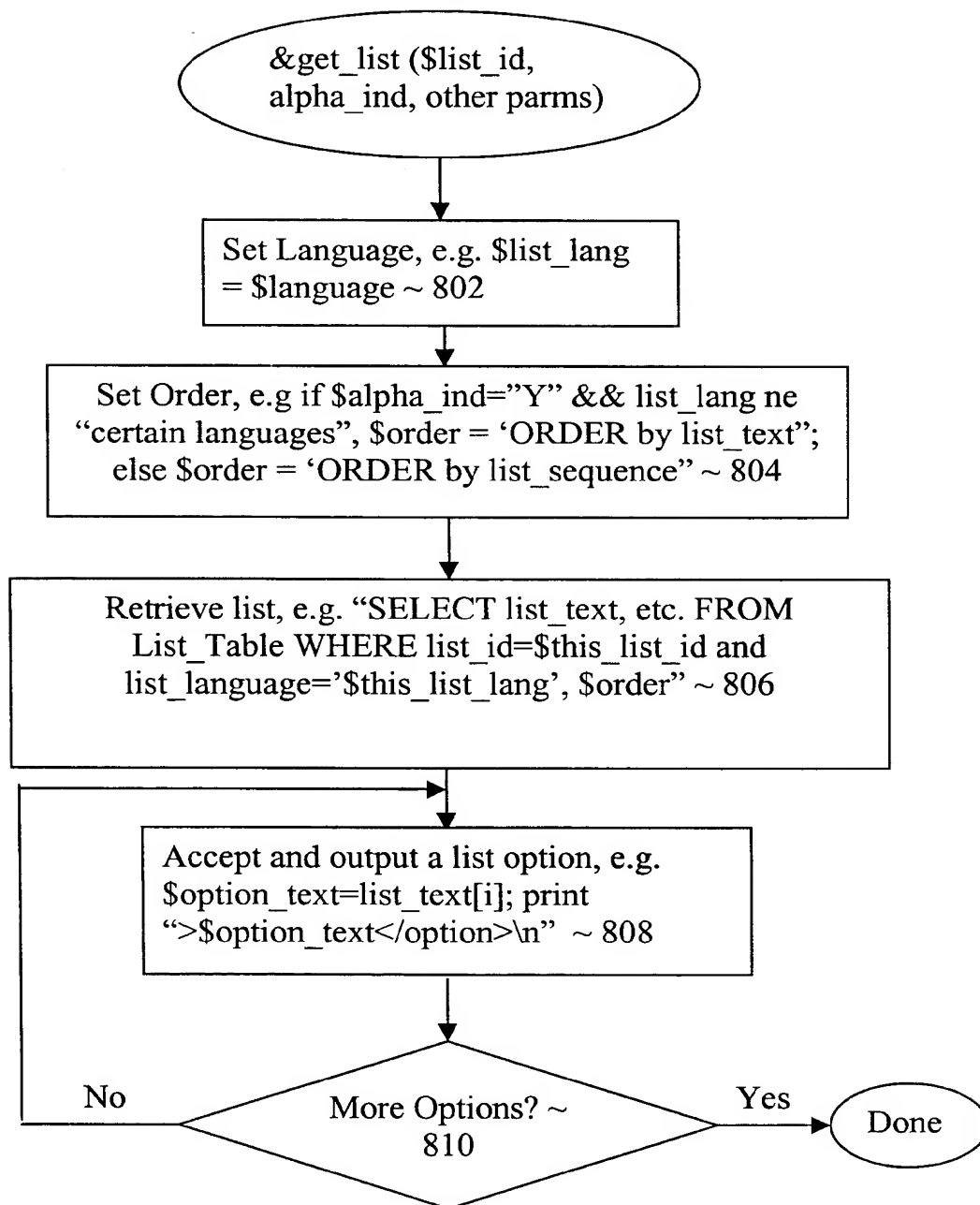
6/13

**Figure 6**

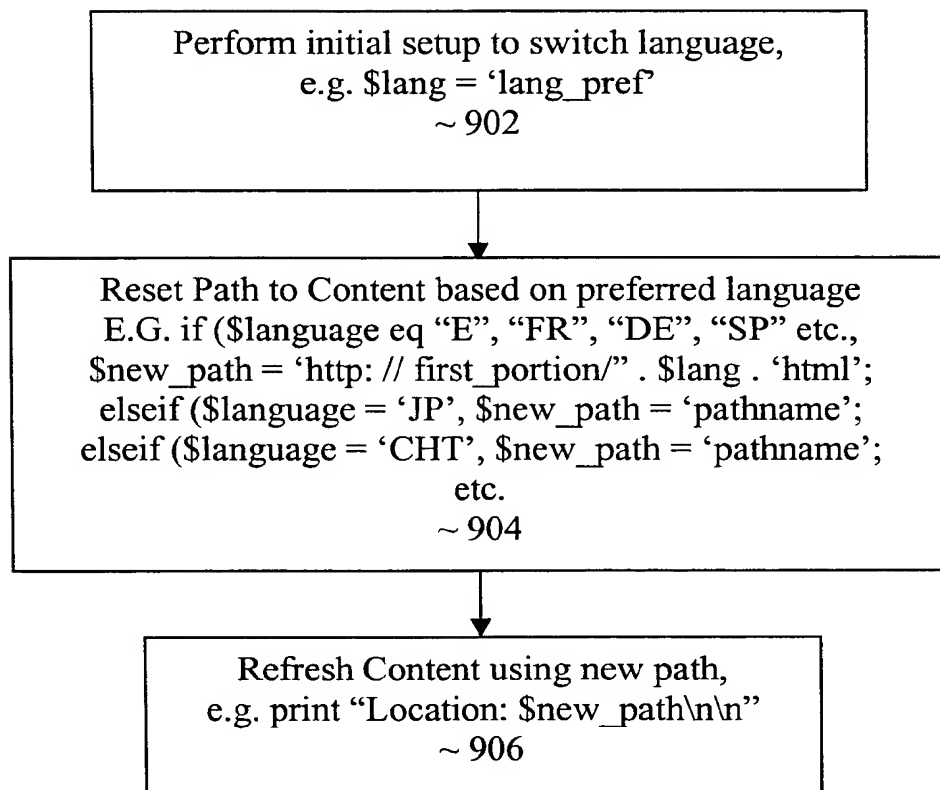
7/13

**Figure 7**

8/13

**Figure 8**

9/13

**Figure 9**

10/13

Message Table ~ 1000

Message ID ~ 1002	Language ID ~ 1004		Message Text ~ 1006	Message Attributes, e.g. font, font size, style, alignment etc. ~ 1008

Figure 10a

List Table ~ 1020

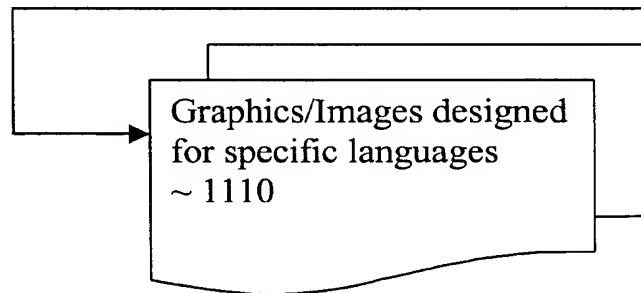
List ID ~ 1022	Language ID ~ 1024	List Text ~ 1026	List Sequence ~ 1028	Other Related Data ~ 1030

Figure 10b

11/13

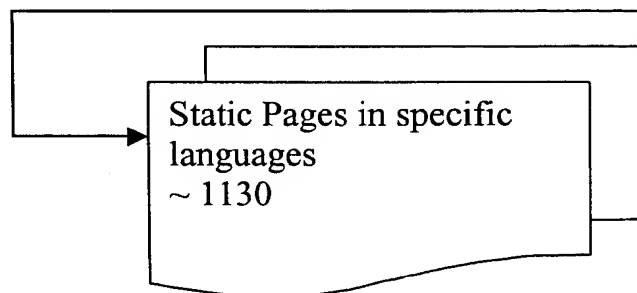
## Graphics/Images Directory ~ 1100

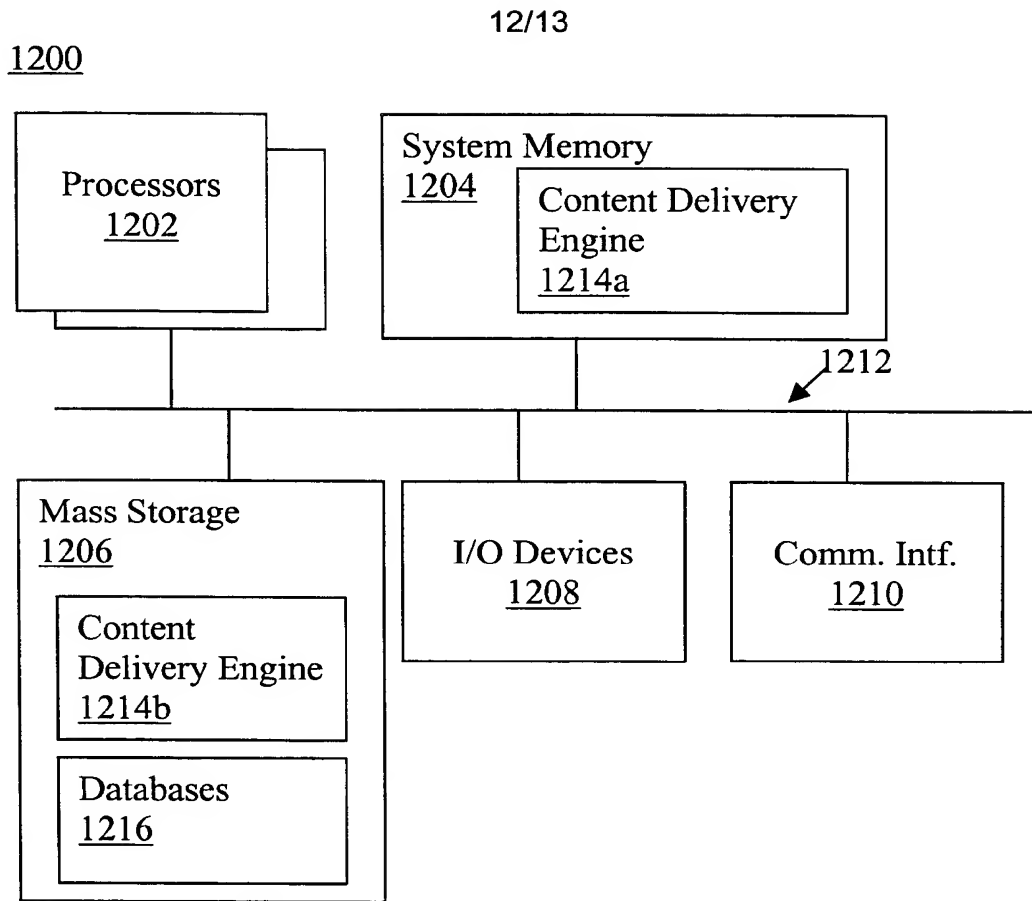
Graphics/Image File Names With Language Based Naming Convention ~ 1102	Graphics/Images File Attributes, e.g. data created, size, etc. ~ 1104	Location ~ 1106

**Figure 11a**

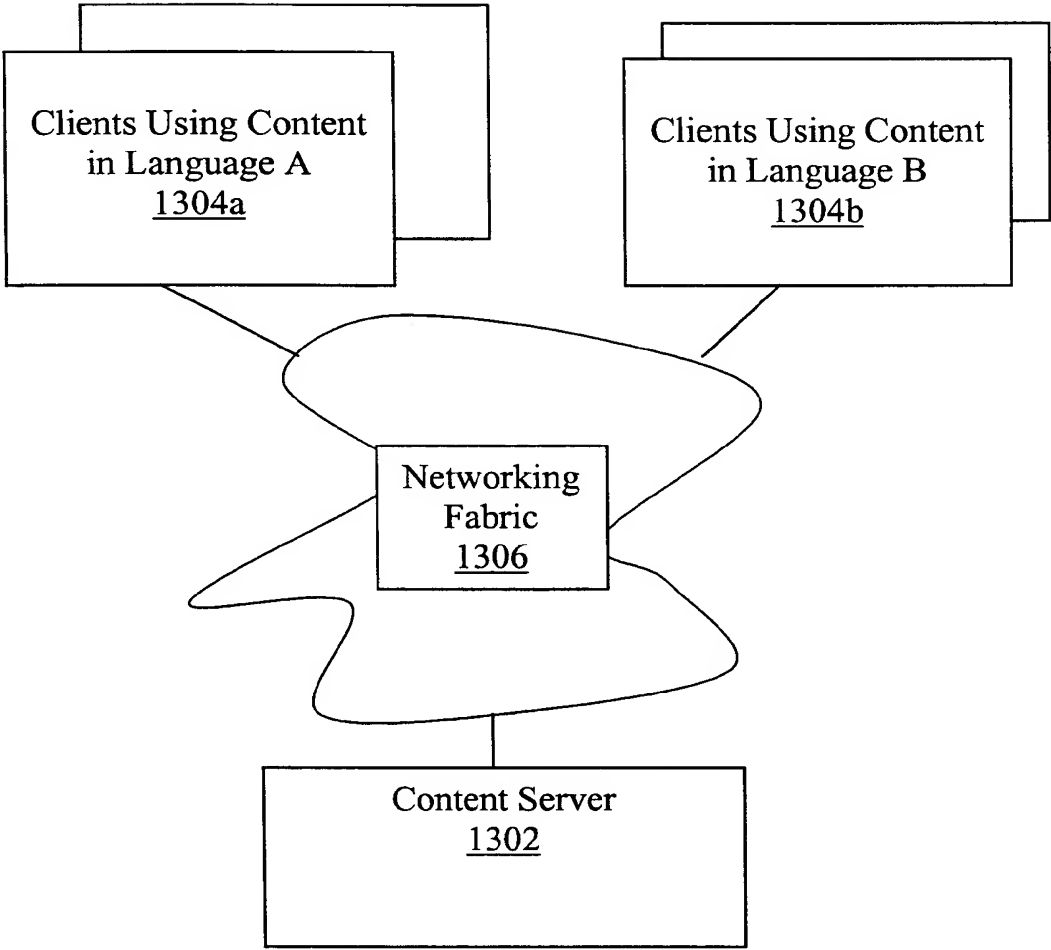
## Static Page Directory ~ 1120

Static Page Names With Language Based Naming Convention ~ 1122	Static Page Attributes, e.g. date created, size, etc. ~ 1124	Location ~ 1126

**Figure 11b**

**Figure 12**





1300

**Figure 13**